REMARKS

The Examiner rejected claims 35-48, 50-63, 65-74 and 76-85 under 35 U.S.C. § 103(a) as allegedly being unpatentable over Lawder (US Patent App. Pub. No. 2003/0004938 A1, filed May 7, 2002) in view of Wagner (US Patent App. Pub. No. 2002/0174130 A1, filed March 20, 2001).

The Examiner rejected claims 49, 64, 75 and 86 under 35 U.S.C. § 103(a) as allegedly being unpatentable over Lawder (US Patent App. Pub. No. 2003/0004938 A1, filed May 7, 2002) in view of Wagner (US Patent App. Pub. No. 2002/0174130 A1, filed March 20, 2001), and further in view of Call (US Patent App. Pub. No. 2002/0165707 A1, filed February 26, 2001).

Applicant respectfully traverses the § 103 rejections with the following arguments.

35 U.S.C. § 103(a): Claims 35-48, 50-63, 65-74 and 76-85

The Examiner rejected claims 35-48, 50-63, 65-74 and 76-85 under 35 U.S.C. § 103(a) as allegedly being unpatentable over Lawder (US Patent App. Pub. No. 2003/0004938 A1, filed May 7, 2002) in view of Wagner (US Patent App. Pub. No. 2002/0174130 A1, filed March 20, 2001).

Applicant respectfully contends that claims 35, 50, 65, and 76 are not unpatentable over Lawder in view of Wagner, because Lawder in view of Wagner does not teach or suggest each and every feature of claims 35, 50, 65, and 76.

As a first example of why claims 35, 50, 65, and 76 are not unpatentable over Lawder in view of Wagner, Lawder in view of Wagner does not teach or suggest the feature: "said executing said algorithm comprising sorting S sequences of binary bits in ascending or descending order of a value associated with each sequence and in a time period denoted as a sorting execution time, said S sequences being stored in a memory device of the computer system prior to said sorting, S being at least 2".

The Examiner argues: "Lawder discloses a computer program product, comprising: a computer usable medium having a computer readable program code embodied therein, said computer readable program code comprising an algorithm for sorting S input sequences of binary bits of a value associated with each sequence (Page 7, [0103], lines 1 – 8, Lawder) said S sequences being stored in a memory device of a computer system prior to said sorting, S being at least 2 (Page 2, [0024], lines 1 – 5, Lawder)".

In response, Applicant respectfully contends that the Examiner is incorrect in alleging that Lawder, page 2, Par. 24 discloses S sequences of binary bits being stored in a memory device of a

computer system prior to being sorted. In fact, Applicant cannot find a disclosure anywhere in Lawder of performing a sorting operation in conjunction with Lawder's invention.

Therefore, Applicant respectfully requests that the Examiner identify with specificity and clarity, in Lawder, sorted sequences of binary bits that were allegedly stored in a memory device of a computer system prior to being sorted, so that Applicant can understand and analyze the Examiner's argument.

As a second example of why claims 35, 50, 65, and 76 are not unpatentable over Lawder in view of Wagner, Lawder in view of Wagner does not teach or suggest the feature: "each sequence of the S sequences comprising K contiguous fields denoted left to right as F_1, F_2, ..., F_K with corresponding field widths of W_1, W_2, ..., W_K".

The Examiner argues: "Lawder discloses ... each sequence of the S sequences comprising K contiguous fields denoted left to right as Fl, F2, . . . , Fk (Fig. 7, Tree level 1, Tree Level 2, Tree Level 3, Page 8, [0123], lines 5 – 7, Lawder) with corresponding field widths of W1, W2, . . . , Wk, (Fig. 7, item: "sub-square sequence numbers (derived-keys)" shows "11", Page 8, [0129], lines 4 – 6, Lawder)."

In response, Applicant interprets the Examiner's argument to allege that each group of contiguous sub-square sequence numbers in each Tree Level (I.e., Tree Levels 1-3) depicted in FIG. 7 of Lawder represent the claimed K contiguous fields. Thus according to the Examiner as best understood by Applicant:

Tree Level 1 has the sequence 00011011 that comprises the contiguous fields "00", "01", "10", 11" (K=4);

Tree Level 2 has 4 identical sequences 00011011, each said sequence comprises the contiguous fields "00", "01", "10", "11" (K=4); and

Tree Level 3 has 4 identical sequences 0001101100011011000110110001101l, each said sequence comprises the contiguous fields "00", "01", "10", "11", "00", "01", "10", "11", "00", "01", "10", "11", "00", "01", "10", "11" (K=16).

Applicant asserts, however, that Lawder does not disclose that said sequences of contiguous fields are stored in a memory device and then sorted, as required by claims 35, 50, 65, and 76 . In fact, Lawder, page 7, Par. 103 specifically states that the tree structure of FIG. 7 of Lawder is a conceptual aid for understanding the mapping process, which implies that the sequences of contiguous fields in the tree structure of FIG. 7 of Lawder are conceptual aids and therefore not real data stored in memory.

Therefore, Applicant respectfully requests that the Examiner identify with specificity and clarity where Lawder allegedly discloses that the sequences of contiguous fields in the tree structure of FIG. 7 of Lawder are stored in memory and subsequently sorted.

As a third example of why claims 35, 50, 65, and 76 are not unpatentable over Lawder in view of Wagner, Applicant relates to the following argument by the Examiner: "Lawder is silent with respect to ascending or descending order, and execution time. On the other hand, Wagner discloses: sorting in ascending or descending order of a value associated with each sequence (Page 1, [0011], lines 5 – 8, Wagner) and in a time period denoted as a sorting execution time (Page 4, [0042], lines 1 – 2, Wagner). It would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the Wagner's teachings to the system Lawder. Skilled

artisan would have been motivated to do so, as suggested by Wagner (Page 1, [0008], lines 1 – 4, Wagner), to provide improve the speed and efficiency of the fundamental data operations of computer applications. In addition, both of the references (Lawder and Wagner) teach features that are directed to analogous art and they are directed to the same field of endeavor, such as, databases management systems, sorting, nodes, and trees. This close relation between both of the references highly suggests an expectation of success."

In response, Applicant respectfully contends that the preceding argument by the Examiner for modifying Lawder by the alleged teaching of Wagner is not persuasive, because the Examiner has not indicated specifically which aspect of Lawder would be so modified and how the modification would be implemented for enablement purposes. Additionally, the Examiner has not indicated how such modification of Lawder would "provide improve the speed and efficiency of the fundamental data operations of computer applications". Moreover, the Examiner assertion that such modification of Lawder would "provid[ing] improve the speed and efficiency of the fundamental data operations of computer applications" is ambiguous.

As a fourth example of why claims 35, 50, 65, and 76 are not unpatentable over Lawder in view of Wagner, Lawder in view of Wagner does not teach or suggest that the sorting comprises the features listed in Table 1.

Table 1

designating S memory areas of the memory device as A_1, A_2, , ..., A_S; setting an output index P = 0 and a field index Q = 0; providing a node E having S elements stored therein, said S elements consisting of the S sequences or S pointers respectively pointing to the S sequences; and (Claims 35, 50, 65, 76)
executing program code, including determining a truth or falsity of an assertion that the elements in node E collectively include or point to no more than one unique sequence U of the S sequences, and if said assertion is determined to be false: then generating C child nodes from node E, each child node including all elements in node E having a unique value of field F_{Q+1}, said child nodes denoted as E_0, E_1, ..., E_{C-1} having associated field F_{Q+1} values of V_0, V_1, ..., V_{C-1}, said child nodes E_0, E_1, ..., E_{C-1} being sequenced such that $V_0<V_1< ...<V_{C-1}$, said generating followed by incrementing Q by 1, said incrementing Q followed by iterating from an index I=0 to I=C-1 in steps of 1, wherein iteration I includes setting $E=E_I$ followed by executing the program code recursively at a next level of recursion for the node E; else for each element in node E: incrementing P by 1, next storing in A_P either U or the element pointing to U, and lastly if the program code at all of said levels of recursion has not been fully executed then resuming execution of said program code at the most previous level of recursion at which the program code was partially but not fully executed else exiting the algorithm." (Claims 35 and 50)
counter-controlled looping through program code, said looping including iteratively executing said program code within nested loops, said executing said program code including determining a truth or falsity of an assertion that the elements in node E collectively include or point to no more than one unique sequence U of the S sequences, and if said assertion is determined to be false: then generating C child nodes from node E, each child node including all elements in node E having a unique value of field F_{Q+1}, said child nodes denoted as E_0, E_1, ..., E_{C-1} having associated field F_{Q+1} values of V_0, V_1, ..., V_{C-1}, said child nodes E_0, E_1, ..., E_{C-1} being sequenced such that $V_0<V_1< ...<V_{C-1}$; said generating followed by incrementing Q by 1, said incrementing Q followed by iterating from an index I=0 to I=C-1 in steps of 1, wherein iteration I includes setting $E=E_I$ followed by returning to said counter-controlled looping; else for each element in node E: incrementing P by 1, next storing in A_P either U or the element pointing to U, and lastly if all iterations of said outermost loop have not been executed then returning to said counter-controlled looping else exiting from said algorithm." (Claims 65 and 76)

With respect to the claimed feature of "designating S memory areas of the memory device as A_1, A_2, , ..., A_S", the Examiner argues that Lawder discloses: "designating S memory areas of the memory device as Al, A2, . . . , As (Page 2, [0025], lines 7 – 9 and 13 – 14, data objects are then partitioned into ordered subsets corresponding to pages of storage in a data store... next partition, thus the partitions are ordered, Lawder)".

In response, Applicant notes that the number of memory areas (S) must be equal to the number of sequences (S) being sorted, which the preceding argument by the Examiner has not addressed. Applicant asserts that Lawder does not disclose that the number of pages in the data store is equal to the number (S) of sequences to be sorted.

With respect to the claimed feature of "setting an output index P = 0 and a field index Q = 0", the Examiner argues that Lawder discloses: "setting an output index P=0 and a field index Q=0 (Fig. 10A, 3: next-match <= 0, 4:current-state <= state 0, Page 11, [0181], lines 12 – 14, Lawder)".

In response, Applicant asserts that neither the next-match variable nor the current-state variable represents the output index P, because neither the next-match variable nor the current-state variable is an output index that indexes the memory areas A_1, A_2, , ... in the format A_P.

In further response, Applicant asserts that neither the next-match variable nor the current-state variable represents the field index Q, because neither the next-match variable nor the current-state variable is a field index that indexes the fields F_1, F_2, , ... in the format F_Q.

With respect to the claimed feature of "providing a node E having S elements stored

therein, said S elements consisting of the S sequences or S pointers respectively pointing to the S sequences", the Examiner argues that Lawder discloses: "providing a node E having S elements stored therein, said S elements consisting of the S sequences or S pointers respectively pointing to the S sequences (Fig. 7, "Node(or state)" and "n-points", Page 8, [0134], lines 5 – 11, Lawder; and Page 3, [0032], lines 9 – 14, Wagner... Wherein the node object corresponds to the node E claimed; and the data queue corresponds to the S sequences claimed)".

In response, Applicant asserts that Lawder, page 8, Par. 134, lines 5-11 does not disclose the preceding claimed feature and is totally unrelated to the preceding claimed feature. As to Wagner, Applicant acknowledges that the tree utility data structure in Par. 32 of Wagner comprises the data objects to be sorted which are equivalent to the S sequences to be sorted. However, Par. 35 of Wagner, with reference to FIG. 3 of Wagner, describes how the data objects (i.e., S sequences) are inserted into the tree utility data structure. Applicant asserts that it is clear, from FIG. 3 and the description thereof in Par. 35 of Wagner, that the S sequences are distributed into the nodes of the tree, and the resultant tree utility data structure does not have a node E such that all S sequences are stored in the node E. Therefore, Wagner does not disclose the preceding claimed feature.

In further response, Applicant asserts that the Examiner has not disclosed how Lawder can be modified to include the alleged teaching of Wagner in an enabling manner. In addition, the Examiner has not provided any argument as to why it is allegedly obvious to modify Lawder to include the alleged teaching of Wagner.

Accordingly, the Examiner's citation of Wagner is not persuasive with respect to the preceding claimed feature.

With respect to the claimed feature of: "executing program code, including determining a truth or falsity of an assertion that the elements in node E collectively include or point to no more than one unique sequence U of the S sequences", the Examiner argues that Lawder discloses: "said executing said program code including determining a truth or falsity of an assertion that the elements in node E collectively include or point to no more than one unique sequence U of the S sequences (Fig. 3, item 138, Page 4, [0036], lines 1 – 3, "whether all the characters of the key .. have been read", Wagner)".

In response, Applicant asserts that the Examiner is incorrect in alleging that step 38 in FIG. 3 of Wagner is equivalent to "determining a truth or falsity of an assertion that the elements in node E collectively include or point to no more than one unique sequence U of the S sequences". Applicant directs the Examiner's attention to the flow chart of FIG. 3 of Wagner and the description thereof in Par. 35, which makes it clear that determining whether all characters of a key has been read is equivalent to determining whether all characters of the data object associated with the key have been processed, which is unrelated to whether the elements in the node E collectively include or point to no more than one unique sequence U of the S sequences.

In further response, Applicant asserts that the Examiner has not disclosed how Lawder can be modified to include the alleged teaching of Wagner in an enabling manner. In addition, the Examiner has not provided any argument as to why it is allegedly obvious to modify Lawder to include the alleged teaching of Wagner.

Accordingly, the Examiner's citation of Wagner is not persuasive with respect to the preceding claimed feature.

With respect to the claimed feature of "generating C child nodes from node E, each child node including all elements in node E having a unique value of field F_{Q+1}", the Examiner argues that Lawder discloses: "generating C child nodes from node E, each child node including all elements in node E having a unique value of field FQ+1 (Fig. 3, item 150, Page 4, [0036], lines 6 – 8, "create new child node ...", Wagner ... Where the node ID corresponds to the unique value of filed FQ +1 claimed)".

In response, Applicant notes that the fields F are the contiguous fields of each sequence of the S sequence to be sorted (i.e., "each sequence of the S sequences comprising K contiguous fields denoted left to right as F_1, F_2, ..., F_K "). Thus the value of F_{Q+1} is the value of the field of a sequence, said field identified by the integer index Q+1. In contrast, the node ID identifies a node of Wagner's tree utility data structure and therefore does not identify the value of the field of a sequence.

In further response, Applicant asserts that the Examiner has not disclosed how Lawder can be modified to include the alleged teaching of Wagner in an enabling manner. In addition, the Examiner has not provided any argument as to why it is allegedly obvious to modify Lawder to include the alleged teaching of Wagner.

Accordingly, the Examiner's citation of Wagner is not persuasive with respect to the preceding claimed feature.

With respect to the claimed feature of "said child nodes denoted as E_0, E_1, ..., E_{C-1} having associated field F_{Q+1} values of V_0, V_1, ..., V_{C-1}, said child nodes E_0, E_1, ..., E_{C-1} being sequenced such that $V_0 < V_1 < ... < V_{C-1}$ ", the Examiner argues that Lawder discloses: "said child nodes denoted

as E0, El,. . . , EC-1 having associated field FQ+1 values of V0, VI, .. . , VC-1, said child nodes E0, El,. . . , EC-1 being sequenced such that V0<V1 < ... <VC-1 Page 3, [0032], lines 9 – 14, Wagner)".

In response, Applicant notes that Wagner, page 3, Par. 32, lines 4-14 recites: "In the example depicted in FIGS. 2A-2D, five additional data objects (zebra, lion, squirrel, mouse, and peacock) are associated with five keys (2, 13, 17, 200, and 2003). The sixth data object (zebra) is stored in the data queue associated with the node object 70. The data flag property of the node object 70 is flagged and is associated with the sixth key (2). As can be appreciated from the preceding insertion, the key can be of variable length. The seventh data object (lion) is stored in the data queue property associated with the node object 84. The data flag property of the node object 84 is flagged and is associated with the seventh key (13)."

Applicant asserts that the preceding quote from Par. 32 of Wagner does not disclose the preceding claimed feature. For example, Wagner clearly does not disclose the ordering relationships expressed as "said child nodes E_0, E_1, ..., E_{C-1} being sequenced such that $V_0<V_1< ...<V_{C-1}$".

In further response, Applicant asserts that the Examiner has not disclosed how Lawder can be modified to include the alleged teaching of Wagner in an enabling manner. In addition, the Examiner has not provided any argument as to why it is allegedly obvious to modify Lawder to include the alleged teaching of Wagner.

Accordingly, the Examiner's citation of Wagner is not persuasive with respect to the preceding claimed feature.

With respect to the claimed feature of:

> "said generating followed by incrementing Q by 1, said incrementing Q followed by iterating from an index I=0 to I=C-1 in steps of 1, wherein iteration I includes setting $E=E_I$ ***followed by executing the program code recursively at a next level of recursion for the node E***" (claims 35 and 50) and
>
> "said generating followed by incrementing Q by 1, said incrementing Q followed by iterating from an index I=0 to I=C-1 in steps of 1, wherein iteration I includes setting $E=E_I$ ***followed by returning to said counter-controlled looping***" (claims 65 and 76),

the Examiner argues that Lawder discloses: "said generating followed by incrementing Q by 1 (Fig. 10A, item 24: current-tree-level<=current-tree-level + 1, Lawder), said incrementing Q followed by iterating from an index 1=0 to 1=C-1 in steps of 1, wherein iteration I includes setting E=EI followed by returning to said counter-controlled looping (Page 4, [0036], lines 8 – 10 and 17 – 19, "current_node is set to the new child node object", Wagner)".

In response, Applicant asserts that none of the preceding citations to Lawder and Wagner disclose "said generating followed by incrementing Q by 1, said incrementing Q followed by iterating from an index I=0 to I=C-1 in steps of 1, wherein iteration I includes setting $E=E_I$ ".

Furthermore, claims 35 and 50 implement the sorting by recursion ("followed by executing the program code recursively at a next level of recursion for the node E"), and claims 75 and 74 implement the sorting by counter-controlled looping ("followed by returning to said counter-controlled looping"). Applicant maintains that the Examiner's citations do not disclose algorithmic implementation by either recursion or counter-controlled looping. Moreover, the

Examiner has not even addressed the recursion embodiment ("followed by executing the program code recursively at a next level of recursion for the node E").

With respect to the claimed feature of "else for each element in node E: incrementing P by 1, next storing in A_P either U or the element pointing to U", the Examiner argues that Lawder discloses: "for each element in node E: incrementing P by 1 (Fig. 4, item 166, Page 4, [0036], lines 16 – 19, "the data flag of the node object is flagged", Wagner), next storing in AP either U or the element pointing to U (Fig. 3, item 162, Page 4, [0036], lines 14 – 17, the data object is pushed in to the data queue of the node object...", Wagner)".

In response, Applicant asserts that the operation described by "the data flag of the node object is flagged" is not equivalent to the operation of "for each element in node E: incrementing P by 1". In addition, Applicant asserts that the data queue of the node object (into which the data object is pushed) does not correspond to the memory area A_P indexed by the output index P.

In further response, Applicant asserts that the Examiner has not disclosed how Lawder can be modified to include the alleged teaching of Wagner in an enabling manner. In addition, the Examiner has not provided any argument as to why it is allegedly obvious to modify Lawder to include the alleged teaching of Wagner.

Accordingly, the Examiner's citation of Wagner is not persuasive with respect to the preceding claimed feature.

With respect to the claimed feature of:

> "lastly if the program code at all of said levels of recursion has not been fully executed then resuming execution of said program code at the most previous level of recursion at which the program code was partially but not fully executed else exiting the algorithm" (claims 35 and 50), and
>
> "lastly if all iterations of said outermost loop have not been executed then returning to said counter-controlled looping else exiting from said algorithm."(claims 65 and 76),

the Examiner argues that Lawder discloses: "lastly if all iterations of said outermost loop have not been executed then returning to said counter-controlled looping else exiting from said algorithm (Page 4, [0036], lines 17 – 19, Wagner)."

In further response, Applicant asserts that claims 35 and 50 implement the sorting by recursion and claims 75 and 74 implement the sorting by counter-controlled looping, as reflected in the preceding claimed features. Applicant maintains that the Examiner's citation to Par. 36 of Wagner does not disclose algorithmic implementation by either recursion or counter-controlled looping. Moreover, the Examiner has not even addressed the recursion embodiment ("lastly if the program code at all of said levels of recursion has not been fully executed then resuming execution of said program code at the most previous level of recursion at which the program code was partially but not fully executed else exiting the algorithm").

As a general remark, Applicant notes that claims 35, 50, 65, and 76 are claiming steps of a sorting algorithm. However, Lawder does not disclose a sorting algorithm but rather discloses an

algorithm for mapping multidimensional data into a Hilbert Curve in a one-dimensional form. In contrast, Wagner discloses a sorting algorithm. Moreover, the algorithms in Lawder and Wagner are significantly different from each other both functionally (i.e., data mapping versus data sorting) and logically. Therefore, modifying Lawder by the alleged teachings of Wagner is not only unobvious because of their functional and logical differences, but also because so modifying Lawder by the alleged teachings of Wagner is very difficult, if not impossible, and is thus not enabled.

Based on the preceding arguments, Applicant respectfully maintains that claim 35, 50, 65, and 76 are not unpatentable over Lawder in view of Wagner, and that claim 35, 50, 65, and 76 are in condition for allowance. Since claims 36-48 depend from claim 35, Applicant contends that claims 36-48 are likewise in condition for allowance. Since claims 51-64 depend from claim 50, Applicant contends that claims 51-64 are likewise in condition for allowance. Since claims 66-74 depend from claim 65, Applicant contends that claims 66-74 are likewise in condition for allowance. Since claims 77-85 depend from claim 76, Applicant contends that claims 77-85 are likewise in condition for allowance.

In addition with respect to claims 36, 51, 66, and 77, Lawder in view of Wagner does not disclose the feature: "wherein said sorting does not include comparing a value of a first sequence of the S sequences with a value of a second sequence of the S sequences".

The Examiner argues that the preceding claimed feature is disclosed in Wagner, page 2, Par. 22, lines 4-9 in combination with the following allegation of Examiner: "Wherein the step of

utilizing a sort algorithm by preserving the initial order of items with equal keys corresponds to the step of utilizing an algorithm that is not adapted to execute comparing as claimed".

In response, Applicant asserts that preserving the initial order of items with equal keys does not imply that there is no comparing as claimed during performance of the sorting in Wagner. Moreover, Wagner does not anywhere disclose the limitation of not using any compare operation on the sequences to be sorted during sorting.

In addition, even if Wagner is not adapted to not perform the claimed comparing during sorting, it does not follow that Lawder's algorithm does not include the claimed comparing during sorting.

Therefore, the Examiner has not established a *prima facie* case of obviousness in relation to claims 36, 51, 66, and 77.

In addition with respect to claims 37, 52, 67, and 78, Lawder in view of Wagner does not disclose the feature: "wherein the sorting execution time is a linear function of a sequence length comprised by each sequence of the S sequences".

The Examiner argues that the preceding claimed feature is disclosed in Wagner, page 2, Par. 22, lines 4-6.

In response, Applicant notes that Wagner, page 2, Par. 22, lines 4-6 recite: "This constant bound on the depth of the tree allows the implementation of a linear sort algorithm". However, Wagner does not define a linear sort algorithm to have the characteristic of "the sorting execution time is a linear function of a sequence length comprised by each sequence of the S sequences"..

In addition, that fact that Wagner's algorithm is a linear sort algorithm does not imply that

Lawder's algorithm is a linear sort algorithm. Thus, Lawder in view of Wagner does not disclose the preceding claimed feature irrespective of the meaning of "linear sort algorithm" in Wagner.

Therefore, the Examiner has not established a *prima facie* case of obviousness in relation to claims 37, 52, 67, and 78.

In addition with respect to claims 38, 53, 68, and 79, Lawder in view of Wagner does not disclose the feature: "wherein the sorting execution time is a linear or less than linear function of S".

The Examiner argues that the preceding claimed feature is disclosed in Wagner, page 2, Par. 22, lines 4-6 and 13-15.

In response, Applicant notes that Wagner, page 2, Par. 22, lines 4-6 and 13-15 recite: ""This constant bound on the depth of the tree allows the implementation of a linear sort algorithm ... However, as n grows without bound, duplicates will be encountered when the key word length is less than log n".

In response, Applicant asserts that the preceding citation to Wagner does not disclose that "the sorting execution time is a linear or less than linear function of S".

In addition, even if in Wagner the sorting execution time is a linear or less than linear function of S, it does not follow that in Lawder's algorithm the sorting execution time is a linear or less than linear function of S.

Therefore, the Examiner has not established a *prima facie* case of obviousness in relation to claims 38, 53, 68, and 79.

In addition with respect to claims 39, 54, 69, and 80, Lawder in view of Wagner does not disclose the feature: "wherein the sorting execution time is essentially independent of an extent to which the S sequences are ordered in the memory device, prior to said sorting, with respect to said associated values".

The Examiner argues that the preceding claimed feature is disclosed in Wagner, page 2, Par. 22, lines 12-13.

In response, Applicant notes that Wagner, page 2, Par. 22, lines 11-13 recite: "the sort algorithms based on the string tree utility data structure do not pose a limit on the number of keys n".

Applicant asserts that the preceding citation to Wagner has no relevance to the preceding claimed feature.

Moreover, even if the preceding claimed feature applies to Wagner, it does not follow that the preceding claimed feature applies to Lawder.

Therefore, the Examiner has not established a *prima facie* case of obviousness in relation to claims 39, 54, 69, and 80.

In addition with respect to claims 40, 55, 70, and 81, Lawder in view of Wagner does not disclose the feature: "wherein the sorting execution time is a decreasing function of a data density of the S sequences".

The Examiner argues that the preceding claimed feature is disclosed in Wagner, page 2, Par. 22, line 15 "less than log n".

In response, Applicant notes that Wagner, page 2, Par. 22, lines 14-15 recite: "as n grows

without bound, duplicates will be encountered when the key word length is less than log n", which has no relevance to the preceding claimed feature.

Moreover, even if the preceding claimed feature applies to Wagner, it does not follow that the preceding claimed feature applies to Lawder.

Therefore, the Examiner has not established a *prima facie* case of obviousness in relation to claims 40, 55, 70, and 81.

In addition with respect to claims 41 and 56, Lawder in view of Wagner does not disclose the feature: "wherein the S elements consist of the S pointers, wherein the node E having the S sequences prior to said sorting includes a linked list that comprises the S pointers, and wherein each child node having pointers therein includes a linked list that comprises said pointers therein".

The Examiner argues that the preceding claimed feature is disclosed in Lawder, page 8, Par. 123, lines 5-10.

In response, Applicant notes that Lawder, page 8, Par. 123, lines 5-10 recite: "Each node corresponds to a first order curve and a collection of nodes at any tree level, k, describes a curve of order k, where the root resides at level 1. Thus the root node corresponds to the first order curve of FIG. 4 and the leaf nodes correspond to the set of first order curves comprising the third order curve of FIG. 6."

Applicants asserts that the preceding citation to Lawder does not disclose the use of a linked list that comprises the S pointers, prior to said sorting.

Therefore, the Examiner has not established a *prima facie* case of obviousness in relation to claims 41 and 56.

In addition with respect to claims 42, 57, 71 and 82, Lawder in view of Wagner does not disclose the feature: "wherein the S sequences each represent a variable-length character string, wherein each of the S character strings consists of the K contiguous fields, wherein K is a sequence-dependent variable subject to $W_1=W_2= \ldots =W_K$ = one byte consisting of a fixed number of binary bits for representing one character".

The Examiner argues: "the combination of Lawder in view of Wagner discloses a computer program product, wherein the S sequences each represent a variable-length character string (Page 11, [0178], lines 1 -4,"bit-string", Lawder), wherein each of the S character strings consists of the K contiguous fields (Fig. 7, Tree level 1, Tree Level 2, Tree Level 3, Page 8 and 11, [0123] and [0178], lines5 – 7 and 2 – 3; respectively, Lawder), wherein K is a sequence-dependent variable subject to W1=W2= . . . =WK=one byte consisting of a fixed number of binary bits for representing one character (Fig. 7, item: "sub-square sequence numbers (derived-keys)" shows "11", Page 8, [0129], lines 4 – 6, Lawder)."

In response, Applicant notes that the bit string in Lawder, page 11, Par, 178, lines 1-4 has nk bits, wherein n is the number of dimensions and k is the order of the Hilbert curve. Thus, both n and k are constant (i.e, fixed length), so that the bit strings are not of variable length.

In further response, Applicant asserts that there is no disclosure in Lawder that the sub-square sequence numbers represent character strings. Furthermore, the sub-square sequence numbers are not the sequences to be sorted and are not comprised by the sequences to be sorted.

Therefore, the Examiner has not established a *prima facie* case of obviousness in relation to claims 42, 57, 71 and 82.

In addition with respect to claims 43, 58, 72, and 83, Lawder in view of Wagner does not disclose the feature: “wherein the S sequences each represent a fixed-length character string, wherein each of the S character strings consists of the K contiguous fields, wherein K is a sequence-dependent variable subject to $W_1=W_2= \ldots =W_K$ = one byte consisting of a fixed number of binary bits for representing one character”.

The Examiner argues: “the combination of Lawder in view of Wagner discloses a computer program product, wherein the S sequences each represent a fixed-length character string (Page 11, [0178], lines 1 – 4, "bit-string", Lawder... Wherein the nk corresponds to the fixed length claimed), wherein each of the S character strings consists of the K contiguous fields (Fig. 7, Tree level 1, Tree Level 2, Tree Level 3, Page 8 and 11, [0123] and [0178], lines5 – 7 and 2 – 3; respectively, Lawder), wherein K is a sequence-dependent variable subject to W1=W2= . . . =WK=one byte consisting of a fixed number of binary bits for representing one character (Fig. 7, item: "sub-square sequence numbers (derived-keys)" shows "11", Page 8, [0129], lines 4 – 6, Lawder).”

In response, Applicant asserts that there is no disclosure in Lawder that the sub-square sequence numbers represent character strings. Furthermore, the sub-square sequence numbers are not the sequences to be sorted and are not comprised by the sequences to be sorted

Therefore, the Examiner has not established a *prima facie* case of obviousness in relation to claims 43, 58, 72, and 83.

In addition with respect to claims 44 and 59, Lawder in view of Wagner does not disclose the feature: “wherein the S sequences consist of S fixed-length words such that each of the S

words has N binary bits, wherein N is at least 2".

The Examiner argues that the preceding claimed feature is disclosed in Lawder, page 8, Par. 129, lines 4-6 and Wagner, page 2, Par. 24, lines 1-3.

In response, Applicant notes that Lawder, page 8, Par. 129, lines 4-6 (" This places the point in the quadrant number is 11. The derived-key of P is now `101") discusses derived keys which are not the sequences to be sorted.

In further response, Applicant asserts that Wagner, page 2, Par. 24, lines 1-3 ("When performing the fundamental data operations, the data manager 14 employs the string tree utility data structure according to the present invention.") is totally irrelevant to the preceding claimed feature.

Therefore, the Examiner has not established a *prima facie* case of obviousness in relation to claims 44 and 59.

In addition with respect to claims 45 and 60, Lawder in view of Wagner does not disclose the feature: "wherein the S words each represent an integer".

The Examiner argues that the preceding claimed feature is disclosed in Lawder, page 5 Par. 76, lines 9-15 and Wagner, page 3, Par. 34, lines 5-9.

In response, Applicant asserts that the "integer value" discussed in Lawder, page 5 Par. 76, lines 9-15 is not an integer value of a word to be sorted.

In response, Applicant asserts that the "integer keys" discussed in Wagner, page 3, Par. 34, lines 5-9 are not integers to be sorted.

Therefore, the Examiner has not established a *prima facie* case of obviousness in relation

to claims 45 and 60.

In addition with respect to claims 46 and 61, Lawder in view of Wagner does not disclose the feature: " wherein the method further comprises determining a leftmost significant bit position of the S words collectively, and wherein the leftmost bit position of field F_1 is the leftmost significant bit position of the S words collectively".

The Examiner argues that the preceding claimed feature is disclosed in Lawder, page 8, Par. 128, lines 1-3 and page 12, Par. 192, lines 1-3.

In response, Applicant asserts that the preceding citations to Lawder do not relate to "a leftmost significant bit position of the S words collectively".

Therefore, the Examiner has not established a *prima facie* case of obviousness in relation to claims 46 and 61.

In addition with respect to claims 47 and 62, Lawder in view of Wagner does not disclose the feature: "wherein the S words each represent a floating point number having the following fields contiguously ordered from left to right: a sign field, an exponent field, and a mantissa field".

The Examiner argues that the preceding claimed feature is disclosed in Lawder, page 8, Par. 135, lines 1-3.

In response, Applicant asserts that the citation to Lawder, page 8, Par. 135, lines 1-3 ("The minimum number of states, or orientations of first order curves, required to encapsulate the Hilbert Curve of any order is given by the expression 2^{n-1}. ") is totally unrelated to the S words each representing a floating point number.

Therefore, the Examiner has not established a *prima facie* case of obviousness in relation to claims 47 and 62.

In addition with respect to claims 48 and 63, Lawder in view of Wagner does not disclose the feature: "wherein generating the C child nodes from node E comprises performing (M AND X) or (X AND M) with a mask M for each sequence X in node E, wherein the mask M is keyed to the field F_{Q+1}, and wherein the bit positions of the mask M relating to the field F_{Q+1} each have a 1 bit, and wherein the remaining bit positions of the mask M each have a 0 bit".

The Examiner argues that the preceding claimed feature is disclosed in Lawder, page 17, lines 15-33.

In response, Applicant notes that the citation to Lawder, page 17, lines 15-33 does not pertain to "the bit positions of the mask M relating to the field F_{Q+1}".

Moreover, Applicant cannot find in Lawder, page 17, lines 15-33, any performance of (M AND X) or (X AND M) "with a mask M for each sequence X in node E".

Therefore, the Examiner has not established a *prima facie* case of obviousness in relation to claims 48 and 63.

In addition with respect to claims 73 and 84, Lawder in view of Wagner does not disclose the feature: "wherein the S sequences consist of S fixed-length integers such that each of the S integers has N binary bits, wherein N is at least 2".

The Examiner argues that the preceding claimed feature is disclosed in Lawder, page 8, Par. 129, lines 4-6.

In response, Applicant notes that Lawder, page 8, Par. 129, lines 4-6 (" This places the point in the quadrant number is 11. The derived-key of P is now `101") discusses derived keys which are not the sequences to be sorted.

Therefore, the Examiner has not established a *prima facie* case of obviousness in relation to claims 73 and 84.

In addition with respect to claims 74 and 85, Lawder in view of Wagner does not disclose the feature: "wherein the S sequences consist of S fixed-length floating point numbers, each of said floating point numbers having the following fields contiguously ordered from left to right: a sign field, an exponent field, and a mantissa field.".

The Examiner argues that the preceding claimed feature is disclosed in Lawder, page 8, Par. 135, lines 1-3.

In response, Applicant assets that the citation to Lawder, page 8, Par. 135, lines 1-3 ("The minimum number of states, or orientations of first order curves, required to encapsulate the Hilbert Curve of any order is given by the expression 2^{n-1}. ") is totally unrelated to the S words each representing a floating point number.

Therefore, the Examiner has not established a *prima facie* case of obviousness in relation to claims 74 and 85.

35 U.S.C. § 103(a): Claims 49, 64, 75, and 86

The Examiner rejected claims 49, 64, 75, and 86 under 35 U.S.C. § 103(a) as allegedly being unpatentable over Lawder (US Patent App. Pub. No. 2003/0004938 A1, filed May 7, 2002) in view of Wagner (US Patent App. Pub. No. 2002/0174130 A1, filed March 20, 2001), and further in view of Call (US Patent App. Pub. No. 2002/0165707 A1, filed February 26, 2001).

Since claims 36-48 depend from claim 35, which Applicants have argued *supra* to not be unpatentable over Lawder in view of Wagner under 35 U.S.C. §103(a), Applicant maintains that claims 36-48 are likewise not unpatentable over Lawder in view of Wagner and further in view of Call under 35 U.S.C. §103(a).

Since claims 51-63 depend from claim 50, which Applicants have argued *supra* to not be unpatentable over Lawder in view of Wagner under 35 U.S.C. §103(a), Applicant maintains that claims 51-63 are likewise not unpatentable over Lawder in view of Wagner and further in view of Call under 35 U.S.C. §103(a).

Since claims 66-74 depend from claim 65, which Applicants have argued *supra* to not be unpatentable over Lawder in view of Wagner under 35 U.S.C. §103(a), Applicant maintains that claims 66-74 are likewise not unpatentable over Lawder in view of Wagner and further in view of Call under 35 U.S.C. §103(a).

Since claims 77-85 depend from claim 76, which Applicants have argued *supra* to not be unpatentable over Lawder in view of Wagner under 35 U.S.C. §103(a), Applicant maintains that claims 77-85 are likewise not unpatentable over Lawder in view of Wagner and further in view of Call under 35 U.S.C. §103(a).

In addition with respect to claims 49, 64, 75, and 86, Lawder in view of Wagner and further in view of Call does not disclose the feature: "wherein W_1, W_2, ..., W_K is such that the sorting execution time is less than a Quicksort execution time for sorting the S sequences via execution of a Quicksort sorting algorithm by said processor".

The Examiner alleges that Call, page 6, Par. 69, lines 10-15 discloses the preceding claimed feature,

In response, Applicant asserts that Call, page 6, Par. 69, lines 10-15 merely describes the external compare function types of QuickSort. Clearly, Call, page 6, Par. 69, lines 10-15 does not disclose that a sorting execution time that is less than a Quicksort execution time for sorting the S sequences.

Moreover, even if Call discloses a sorting execution time that is less than a Quicksort execution time for sorting the S sequences, there is no way to modify Lawder and Quicksort to likewise achieve a sorting execution time that is less than a Quicksort execution time for sorting the S sequences, other than to replace Lawder and Quicksort by the algorithm of Call.

Therefore, the Examiner has not established a *prima facie* case of obviousness in relation to claims 49, 64, 75, and 86.

CONCLUSION

Based on the preceding arguments, Applicant respectfully believes that all pending claims and the entire application meet the acceptance criteria for allowance and therefore request favorable action. If the Examiner believes that anything further would be helpful to place the application in better condition for allowance, Applicant invites the Examiner to contact Applicant's representative at the telephone number listed below. The Director is hereby authorized to charge and/or credit Deposit Account No. 09-0457.

Date: 12/01/2006

Jack P. Friedman

Jack P. Friedman
Registration No. 44,688

Schmeiser, Olsen & Watts
22 Century Hill Drive - Suite 302
Latham, New York 12110
(518) 220-1850